



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

+source +code +format +editor +change +block +text


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Published before December 2003

Terms used: [source](#) [code](#) [format](#) [editor](#) [change](#) [block](#) [text](#)

Found 1,091 of 150,460

Sort results
by

relevance

[Save results to a Binder](#)[Try an Advanced Search](#)Display
results

expanded form

[Search Tips](#)Try this search in [The ACM Guide](#)
☐ Open results in a new
window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐**1** [Compiler construction: an advanced course](#)

F. L. Bauer, F. L. De Remer, M. Griffiths, U. Hill, J. J. Horning, C. H. A. Koster, W. M. McKeeman, P. C. Poole, W. M. Waite, G. Goos, J. Hartmanis
January 1974 Book

Publisher: Springer-Verlag New York, Inc.Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#)

The Advanced Course took place from March 4 to 15, 1974 and was organized by the Mathematical Institute of the Technical University of Munich and the Leibniz Computing Center of the Bavarian Academy of Sciences, in co-operation with the European Communities, sponsored by the Ministry for Research and Technology of the Federal Republic of Germany and by the European Research Office, London.


2 [Fast detection of communication patterns in distributed executions](#)

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research CASCON '97**

Publisher: IBM PressFull text available:  [pdf\(4.21 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

3 [Proceedings of the SIGNUM conference on the programming environment for development of numerical software](#)March 1979 **ACM SIGNUM Newsletter**, Volume 14 Issue 1**Publisher:** ACM PressFull text available:  [pdf\(5.02 MB\)](#) Additional Information: [full citation](#)

4 Classics in software engineering

January 1979 Divisible Book

Publisher: Yourdon PressAdditional Information: [full citation](#), [cited by](#), [index terms](#)**5 Anatomy of LISP**

John Allen

January 1978 Book

Publisher: McGraw-Hill, Inc.Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

This text is nominally about LISP and data structures. However, in the process it covers much broader areas of computer science. The author has long felt that the beginning student of computer science has been getting 'a distorted and disjointed picture of the field. In some ways this confusion is natural; the field has been growing at such a rapid rate that few are prepared to be judged experts in all areas of the discipline. The current alternative seems to be to give a few introductory cou ...

6 Smalltalk-80: the language and its implementation

Adele Goldberg, David Robson

January 1983 Book

Publisher: Addison-Wesley Longman Publishing Co., Inc.Full text available:  [pdf\(33.56 MB\)](#) Additional Information: [full citation](#), [abstract](#), [cited by](#), [index terms](#), [review](#)**From the Preface (See Front Matter for full Preface)**

Advances in the design and production of computer hardware have brought many more people into direct contact with computers. Similar advances in the design and production of computer software are required in order that this increased contact be as rewarding as possible. The Smalltalk-80 system is a result of a decade of research into creating computer software that is appropriate for producing highly functional and interactive ...

7 Essays in computing science

C. A. R. Hoare

January 1989 Book

Publisher: Prentice-Hall, Inc.Full text available:  [pdf\(20.91 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [review](#)

Charles Antony Richard Hoare is one of the most productive and prolific computer scientists. This volume contains a selection of his published papers. There is a need, as in a Shakespearian Chorus, to offer some apology for what the book manifestly fails to achieve. It is not a complete 'collected works'. Selection between papers of this quality is not easy and, given the book's already considerable size, some difficult decisions as to what to omit have had to be made. Pity the editor weighin ...

8 Fortran 8X draft

Loren P. Meissner

December 1989 **ACM SIGPLAN Fortran Forum**, Volume 8 Issue 4**Publisher:** ACM PressFull text available:  [pdf\(21.36 MB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

Standard Programming Language Fortran. This standard specifies the form and establishes the interpretation of programs expressed in the Fortran language. It consists of the specification of the language Fortran. No subsets are specified in this standard. The

previous standard, commonly known as "FORTRAN 77", is entirely contained within this standard, known as "Fortran 8x". Therefore, any standard-conforming FORTRAN 77 program is standard conforming under this standard. New features can b ...

9 The theory of parsing, translation, and compiling

Alfred V. Aho, Jeffrey D. Ullman
January 1972 Book

Publisher: Prentice-Hall, Inc.

Full text available:  pdf(98.28 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

From volume 1 Preface (See Front Matter for full Preface)

This book is intended for a one or two semester course in compiling theory at the senior or graduate level. It is a theoretically oriented treatment of a practical subject. Our motivation for making it so is threefold.


(1) In an area as rapidly changing as Computer Science, sound pedagogy demands that courses emphasize ideas, rather than implementation details. It is our hope that the algorithms and concepts presen ...

10 Interactive Editing Systems: Part II



Norman Meyrowitz, Andries van Dam
September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3

Publisher: ACM Press

Full text available:  pdf(9.17 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

11 Software and document engineering: Supporting document and data views of source code



Michael L. Collard, Jonathan I. Maletic, Andrian Marcus
November 2002 **Proceedings of the 2002 ACM symposium on Document engineering DocEng '02**

Publisher: ACM Press

Full text available:  pdf(162.30 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The paper describes the use of an XML format to store and represent program source code. A new XML application, srcML (SouRCe Markup Language), is presented. srcML presumes a document view of source code where information about the syntactic structure is layered over the original source code document. The resultant multi-layered document has a base layer of all the original text (and formatting). The second layer is the syntactic information, derived from the grammar of the programming language, ...


Keywords: XML, abstract syntax tree, markup language, program analysis, source code

12 On-line Text Editing: A Survey



Andries van Dam, David E. Rice
September 1971 **ACM Computing Surveys (CSUR)**, Volume 3 Issue 3

Publisher: ACM Press

Full text available:  pdf(1.91 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper is a survey of current methods for the on-line creation and editing of computer

programs and of ordinary manuscripts text. The characteristics of on-line editing systems are examined and examples of various implementations are described in three categories: program editors, text editors, and terminals with local editing facilities.

13 Special issue: AI in engineering



D. Sriram, R. Joobbani

April 1985 **ACM SIGART Bulletin**, Issue 92

Publisher: ACM Press

Full text available: pdf(8.79 MB) Additional Information: [full citation](#), [abstract](#)

The papers in this special issue were compiled from responses to the announcement in the July 1984 issue of the SIGART newsletter and notices posted over the ARPAnet. The interest being shown in this area is reflected in the sixty papers received from over six countries. About half the papers were received over the computer network.

14 The object-oriented implementation of a document editor



Paul Calder, Mark Linton

October 1992 **ACM SIGPLAN Notices , conference proceedings on Object-oriented programming systems, languages, and applications OOPSLA '92**, Volume 27 Issue 10

Publisher: ACM Press

Full text available: pdf(1.32 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

15 A structural view of the Cedar programming environment



Daniel C. Swinehart, Polle T. Zellweger, Richard J. Beach, Robert B. Hagmann

August 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 8 Issue 4

Publisher: ACM Press

Full text available: pdf(6.32 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents an overview of the Cedar programming environment, focusing on its overall structure—that is, the major components of Cedar and the way they are organized. Cedar supports the development of programs written in a single programming language, also called Cedar. Its primary purpose is to increase the productivity of programmers whose activities include experimental programming and the development of prototype software systems for a high-performance personal computer. T ...

16 Markup systems and the future of scholarly text processing



James H. Coombs, Allen H. Renear, Steven J. DeRose

November 1987 **Communications of the ACM**, Volume 30 Issue 11

Publisher: ACM Press

Full text available: pdf(1.91 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Markup practices can affect the move toward systems that support scholars in the process of thinking and writing. Whereas procedural and presentational markup systems retard that movement, descriptive markup systems accelerate the pace by simplifying mechanical tasks and allowing the authors to focus their attention on the content.

17 The Desert environment



Steven P. Reiss

October 1999 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 8 Issue 4

Publisher: ACM Press

Full text available:  pdf(868.64 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The Desert software engineering environment is a suite of tools developed to enhance programmer productivity through increased tool integration. It introduces an inexpensive form of data integration to provide additional tool capabilities and information sharing among tools, uses a common editor to give high-quality semantic feedback and to integrate different types of software artifacts, and builds virtual files on demand to address specific tasks. All this is done in an open and extensible ...


Keywords: integrated programming environments, program editors

18 Operating system principles

Per Brinch Hansen

January 1973 Book

Publisher: Prentice-Hall, Inc.

Full text available:  pdf(16.81 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

From the Preface

MAIN GOAL

This book tries to give students of computer science and professional programmers a general understanding of *operating systems*--the programs that enable people to share computers efficiently.


To make the sharing of a computer tolerable, an operating system must enforce certain rules of behavior on all its users. One would therefore expect the designers of operating systems to do their utmost to make them as s ...

19 Cryptography and data security

Dorothy Elizabeth Robling Denning

January 1982 Book

Publisher: Addison-Wesley Longman Publishing Co., Inc.

Full text available:  pdf(19.47 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

From the Preface (See Front Matter for full Preface)

Electronic computers have evolved from exiguous experimental enterprises in the 1940s to prolific practical data processing systems in the 1980s. As we have come to rely on these systems to process and store data, we have also come to wonder about their ability to protect valuable data.


Data security is the science and study of methods of protecting data in computer and communication systems from unauthorized disclosure ...

20 The architecture of concurrent programs

Per Brinch Hansen

January 1977 Book

Publisher: Prentice-Hall, Inc.

Full text available:  pdf(10.71 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

From the Preface**CONCURRENT PROGRAMMING**

This book describes a method for writing concurrent computer programs of high quality. It is written for professional programmers and students who are faced with the complicated task of building reliable computer operating systems or real-time control programs.

The motivations for mastering concurrent programming are both economic and intellectual. Concurrent programming makes it possible to use a compu ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)